

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of: <b>Maison et al.</b>	§	
	§	Group Art Unit: <b>2626</b>
Serial No. <b>10/637,219</b>	§	
	§	Examiner: <b>Neway, Samuel G.</b>
Filed: <b>August 8, 2003</b>	§	
	§	
For: <b>Task Specific Code Generation for</b>	§	
<b>Speech Recognition Decoding</b>	§	

**Commissioner for Patents**  
**P.O. Box 1450**  
**Alexandria, VA 22313-1450**

**35526**  
PATENT TRADEMARK OFFICE  
CUSTOMER NUMBER

**APPEAL BRIEF (37 C.F.R. 41.37)**

This brief is in furtherance of the Notice of Appeal, filed in this case on October 16, 2007.

A fee of \$510.00 is required for filing an Appeal Brief. Please charge this fee to Yee & Associates, P.C. Deposit Account No. 50-3157. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to Yee & Associates, P.C. Deposit Account No. 50-3157.

A 1 month extension of time is believed to be necessary. I authorize the Commissioner to charge the 1 month extension fee of \$120.00 to Yee & Associates, P.C. Deposit Account No. 50-3157. No additional extension of time is believed to be necessary. If, however, an additional extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to Yee & Associates, P.C. Deposit Account No. 50-3157.

**REAL PARTY IN INTEREST**

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

### **RELATED APPEALS AND INTERFERENCES**

This appeal has no related proceedings or interferences.

## **STATUS OF CLAIMS**

### **A. TOTAL NUMBER OF CLAIMS IN APPLICATION**

Claims in the application are: 1-16, 21-24

### **B. STATUS OF ALL THE CLAIMS IN APPLICATION**

Claims canceled: 17-20

Claims withdrawn from consideration but not canceled: none

Claims pending: 1-16, 21-24

Claims allowed: none

Claims rejected: 1-16, 21-24

Claims objected to: none

### **C. CLAIMS ON APPEAL**

The claims on appeal are: 1-16, 21-24

### **STATUS OF AMENDMENTS**

No amendments were submitted after the Final Office Action of July 17, 2007.

## **SUMMARY OF CLAIMED SUBJECT MATTER**

### **A. CLAIM 1 - INDEPENDENT**

The subject matter of claim 1 is directed to a method, implemented in a data processing system, for generating task-specific code for pattern recognition (Specification page 6 line 3 to page 8 line 19, Figure 1 and Figure 2). The method comprises receiving task-specific input system data of a pattern recognition system (Specification page 8 lines 20-24, page 9 lines 1-9, Figure 3, reference numeral 302, 310, Figure 4, reference numeral 402). The method further comprises generating task-specific code for the pattern recognition system based on the task-specific input system data, wherein the task-specific code includes computer language suitable for compilation (Specification page 9 lines 12-18, page 11 lines 13-17, Figure 4 reference numeral 410, 412, Figure 5 reference numeral 502, 504).

### **B. CLAIM 11 – INDEPENDENT**

The subject matter of claim 11 is directed to a computer program product, stored in a computer readable recordable type medium, for generating task-specific code for pattern recognition (Specification page 12 lines 17-19). The computer program product comprises instructions for receiving task-specific input system data of a pattern recognition system, (Specification page 8 lines 20-24, page 9 lines 1-9, Figure 3, reference numeral 302, 310, Figure 4, reference numeral 402) and instructions for generating task-specific code for the pattern recognition system based on the task-specific input system data, wherein the task-specific code includes computer language suitable for compilation (Specification page 9 lines 12-18, page 11 lines 13-17, Figure 4 reference numeral 410, 412, Figure 5 reference numeral 502, 504).

### **C. CLAIM 21 – INDEPENDENT**

The subject matter of claim 21 is directed to an apparatus for generating task-specific code for pattern recognition. The apparatus comprises a bus, a memory connected to the bus, wherein the memory contains computer readable instructions, and a processor connected to the bus (Specification page 6 line 3 to page 8 line 19, Figure 1 and Figure 2). The processor executes the computer readable instructions to receive task-specific input system data of a pattern recognition system, (Specification page 8 lines 20-24, page 9 lines 1-9, Figure 3, reference numeral 302, 310,

Figure 4, reference numeral 402) generate task-specific code for the pattern recognition system based on the task-specific input system data, wherein the task-specific code includes computer language suitable for compilation, and compile the task-specific code to form a decoder program for the pattern recognition system (Specification page 9 lines 12-18, page 11 lines 13-17, Figure 4 reference numeral 410, 412, Figure 5 reference numeral 502, 504).

**D. CLAIM 5 - DEPENDENT**

The subject matter of claim 5 is directed to a method of claim 3, plus the additional features of wherein the language model is represented as a Hidden Markov Model (Specification page 9 line 10).

**E. CLAIM 8 - DEPENDENT**

The subject matter of claim 8 is directed to a method of claim 7, plus the additional features of profiling the decoder program to form a profile and determining whether the decoder program is optimized (Specification page 11 lines 6-25, Figure 4 reference numeral 412, 414, 420, 424, Figure 5 reference numeral 508, 510, 512).

## **GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

The grounds of rejection to review on appeal are as follows:

### **A. GROUND OF REJECTION 1 (CLAIMS 11-16)**

The first ground of rejection is the assertion that claim 11 is directed to non-statutory subject matter under 35 U.S.C. § 101.

### **B. GROUND OF REJECTION 2 (CLAIMS 1-4, 6-7, 11-14, AND 21-23)**

The second ground of rejection is the assertion that claims 1-4, 6-7, 11-14, and 21-23 are obvious under 35 U.S.C. § 103 over *Arnold et al.*, Method and Apparatus for Providing a Dynamic Speech-Driven Control and Remote Service Access System, U.S. Patent Application Publication 2003/0125955, December 28, 2001 (hereinafter “*Arnold*”) in view of *Poirier et al.*, Executable for Requesting a Linguistic Service, U.S. Patent 6,321,372, December 23, 1998 (hereinafter “*Poirier*”).

### **C. GROUND OF REJECTION 3 (CLAIM 5)**

The third ground of rejection is the assertion that claim 5 is obvious under 35 U.S.C. § 103 over *Arnold* in view of *Poirier* and in further view of IBM Technical Disclosure Bulletin, Determining the Probability of Words in a String with a Word-Skipping Model, November 1, 1985 (hereinafter “*IBM Bulletin*”).

### **D. GROUND OF REJECTION 4 (CLAIMS 8-10, 15-16, AND 24)**

The fourth ground of rejection is the assertion that claims 8-10, 15-16, and 24 are obvious under 35 U.S.C. § 103 over *Arnold* in view of *Poirier* and in further view of *Lanning*, Apparatus and Method for Optimizing Applications for Multiple Operational Environments or Modes, U.S. Patent 5,787,285, August 15, 1995 (hereinafter “*Lanning*”).

## ARGUMENT

### **A. GROUND OF REJECTION 1 (CLAIMS 11-16)**

The first ground of rejection is the assertion that claim 11 is directed to non-statutory subject matter under 35 U.S.C. § 101. Claim 11 is a representative claim of this grouping of claims. Claim 11 is as follows:

11. A computer program product, stored in a computer readable recordable type medium, for generating task-specific code for pattern recognition, the computer program product comprising:  
instructions for receiving task-specific input system data of a pattern recognition system; and  
instructions for generating task-specific code for the pattern recognition system based on the task-specific input system data, wherein the task-specific code includes computer language suitable for compilation.

The Final Office Action states the rejection of claim 11 under 35 U.S.C. § 101 as follows:

Claims 11 - 16 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 11 - 16 are directed to a "computer program product".

The "computer program product" can reasonably be interpreted as computer listings only. Computer programs claimed as computer listings per se, i.e., the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory.

Amending the claims to recite '*A recordable type medium storing a computer program product*' would overcome the rejection in accordance with Applicant's disclosure.

Final Office Action of July 17, 2007. (emphasis provided)

The computer program product of the Applicants is stored in a computer readable recordable type medium. Language suggested by the Final Office Action also has a computer program product stored in a recordable type medium. The suggested wording appears to offer a

stylistic alternative only. A search of the USPTO patent database provides over 18,000 patents having program product claims wherein approximately 1000 also have language of “program product stored,” as in the case of the Applicants. Therefore, the Office has as a matter of practice allowed claims having similar language as in the instant claim. In accordance with the accepted practice of the Office, the instant claims are acceptable in language. Accordingly the claims as written comply with 35 U.S.C. § 101.

Additionally, in the case of *In re Beauregard*, the Commissioner of the United States Patent and Trademark Office conceded, in a case before the Federal Circuit, that “computer programs embodied in a tangible medium, such as floppy diskettes, are patentable subject matter under 35 U.S.C. § 101 and must be examined under 35 U.S.C. §§ 102 and 103.” *In re Beauregard*, 35 U.S.P.Q.2d 1383 (Fed. Cir. 1995). Claim 11 provides for a “computer program product comprising... a recordable-type medium,” which, as shown above, is a tangible medium. Therefore, under the standards of *In re Beauregard*, claim 1 is statutory subject matter under 35 U.S.C. § 101. Accordingly, again, this rejection is overcome.

## **B. GROUND OF REJECTION 2 (CLAIMS 1-4, 6-7, 11-14, AND 21-23)**

The second ground of rejection is the assertion that claims 1-4, 6-7, 11-14, and 21-23 are obvious under 35 U.S.C. § 103 over *Arnold et al.*, Method and Apparatus for Providing a Dynamic Speech-Driven Control and Remote Service Access System, U.S. Patent Application Publication 2003/0125955, December 28, 2001 (hereinafter “*Arnold*”) in view of *Poirier et al.*, Executable for Requesting a Linguistic Service, U.S. Patent 6,321,372, December 23, 1998 (hereinafter “*Poirier*”). Claim 1 is a representative claim of this grouping. Claim 1 is as follows:

1. A method, implemented in a data processing system, for generating task-specific code for pattern recognition, the method comprising:  
receiving task-specific input system data of a pattern recognition system; and  
generating task-specific code for the pattern recognition system based on the task-specific input system data, wherein the task-specific code includes computer language suitable for compilation.

The Final Office Action states the rejection of claim 1 under 35 U.S.C. § 103 as follows:

Claim 1: *Arnold* discloses a method, implemented in a data processing system, for generating task-specific code for pattern recognition ([0008]), the method comprising:

receiving task-specific input system data of a pattern recognition system and generating task-specific code for the pattern recognition system based on the task specific input system data ("the distributed speech recognition system allows automatic "speaker adaptation" to be performed locally by the client device...local parameters ... are adapted locally by the client device in performing its speech recognition function", [0008], see also [0009] and "the client device is adapting its models in response to the speaker", [0020]).

However, even if *Arnold* discloses that "the speech recognizer module ... can be represented by one or more software applications", it does not explicitly disclose the software applications being source code.

*Poirier* discloses a similar method where a source code is modified, such as by further specifying it, in a linguistic service system. *Poirier* also discloses the code being compiled ("compile . . . modified source code", col. 10, lines 52-56).

It would have been obvious to one with ordinary skill in the art at the time the invention was made to use a source code (which has to be compiled in order to execute on a computer) as the software modules in *Arnold's* method in order to modify an existent source code in order to generate a more specified service (*Poirier*, col. 2, lines 26-36).

Final Office Action of July 17, 2007.

With regard to claim 1, the cited references, considered as a whole, cannot be used to state a *prima facie* obviousness rejection because the cited combination fails to teach all of the features of claim 1. In particular, the combination of *Arnold* and *Poirier*, considered as a whole, fails to teach the feature of "generating task-specific code for the pattern recognition system based on the task-specific input system data, wherein the task-specific code includes computer language suitable for compilation." The Final Office Action states otherwise asserting various portions of *Arnold* and *Poirier* as follows:

[0008] Specifically, the distributed speech recognition system comprises at least one client device and a central server. The client device is a remote device that is in communication with the central server, but it is physically deployed apart from the central server. In operation, the client device is equipped with a speech recognition module having an initial language model. In one embodiment, the distributed speech recognition system allows automatic "speaker adaptation" to be performed locally by the client device. Namely, local parameters such as environmental noise around the speaker, pronunciation (e.g., accents or dialect) of the speaker and/or the acoustic environment (e.g., within a tunnel or a carpeted room) around the speaker are adapted locally by the client device in performing its speech recognition

functions. Speaker adaptation is particularly appropriate within the present architecture in that it is carried out in a client device largely dedicated to a particular user. Although such adaptations are performed locally, the central server may also assist the client device as necessary, e.g., forwarding a different acoustic model to the client device from the central server.

[0009] For example, in another embodiment, the distributed speech recognition system provides the ability to implement "dynamic grammars". Specifically, the client device is initially equipped with an initial language model. As the user interacts with the client device, the language model is updated by the central server as the interactions between the user and client device indicate that an updated language model is required to carry out the user's request. This distributed approach maximizes the processing power of the client device without overburdening the client device unnecessarily with a complex language model.

...

[0020] In turn, the speech recognizer 120 receives the speech features and is able to decode the "recognized text" from the speech features using various models as discussed below. An important aspect of the present invention pertains to the "dynamic" models that are employed by the speech recognizer 120. Specifically, due to the anticipated small footprint of the client device 110, the present invention employs "dynamic grammars" as a driving mechanism in providing the necessary models or portions or updates of a model to the client device. Since the processing capability and storage capability of the client device 110 are anticipated to be limited, the present invention is designed to provide the client device 110 with just enough data and information to perform the tasks as required by a current speaker. Thus, the client device is adapting its models in response to the speaker, hence the term "dynamic grammars". The speaker adaptation functions are executed in cooperation with the central server 130.

*Arnold*, paragraphs [0008], [0009], and [0020].

The new techniques can also treat result conversion objects hierarchically, allowing a programmer to write source code for a format definition class for a new format and character set based on a hierarchical descendant relationship with an ancestor conversion class for which source code already exists. The ancestor conversion class may, for example, be a less specified class or it may be a proto-class that serves only as an ancestor of one or more conversion classes within a hierarchy. Therefore, source code for the descendant can be produced by modifying the preexisting source code for the ancestor.

...

The operations requested by user signals can include editing operations that modify preexisting source code 100 or intermediate versions to obtain

modified source code 124 and also compile or interpret operations that use modified source code 124 to obtain service executable 126.

*Poirier* col. 2 lines 26-36 and col. 10 lines 52-56.

As in paragraph [0007], *Arnold* is directed to, “a distributed speech recognition system that provides speech driven control and remote services.” Further, *Arnold* teaches performing adaptation on the client device, as shown in paragraph [0008], which provides that “...automatic “speaker adaptation” is to be performed locally by the client device.” *Arnold* further teaches use of minimal resources of the client to perform the adaptation. For example, paragraph [0009] provides that “distributed approach maximizes the processing power of the client device without overburdening the client device unnecessarily with a complex language model.” Additionally, *Arnold* again teaches a client having just the needed resource to perform the task. For example, *Arnold* states that:

An important aspect of the present invention pertains to the “dynamic” models that are employed by the speech recognizer 120. Specifically, due to the anticipated small footprint of the client device 110, the present invention employs “dynamic grammars” as a driving mechanism in providing the necessary models or portions or updates of a model to the client device. Since the processing capability and storage capability of the client device 110 are anticipated to be limited, the present invention is designed to provide the client device 110 with just enough data and information to perform the tasks as required by a current speaker. Thus, the client device is adapting its models in response to the speaker, hence the term “dynamic grammars”. The speaker adaptation functions are executed in cooperation with the central server 130.

*Arnold*, paragraph [0020].

*Arnold* clearly teaches away from heavy processing on the client and towards providing just what is needed when needed at the client. Further, *Arnold* teaches obtaining the ready to use models from the server on an as-needed basis. *Arnold* specifically teaches away from generating code for a client in response to a client request as claimed.

Because *Arnold* teaches away from this claimed feature, no rational underpinning exists to combine the operation of *Arnold* and that of *Poirier*. Therefore, under the standards of *KSR Intl.*, the examiner failed to state a *prima facie* obviousness rejection against the claims.

Furthermore, the teaching of *Arnold* in paragraph [0020] requires “... ready to use models, portions or updates of a model...” for the client to use. The client of *Arnold* works with a granular

technique to maintain a small footprint. Adding the teaching of Poirier to that of Arnold would cause Arnold to become inoperative because a request from a client would no longer be ready to use and require compilation, thereby eviscerating the current method of Arnold.

One of ordinary skill would recognize that combining *Poirier* and *Arnold* would render *Arnold* useless. Thus, one of ordinary skill would avoid combining the references. Hence, no rational underpinning exists to combine the operation of *Arnold* and that of *Poirier*. Therefore, under the standards of *KSR Intl.*, the examiner failed to state a *prima facie* obviousness rejection against the claims.

Still further, the examiner failed to state a proper reason to achieve the legal conclusion of obviousness of claim 1 under the standards of *KSR Intl.* Instead, the examiner only provided a purported advantage to combine the references, which fails to connect that purported advantage to the *legal conclusion* of obviousness, are required by *KSR Int'l.* Instead, the examiner only states that combining the references would be obvious “in order to modify an existent source code in order to generate a more specified service.” Final Office Action of July 17, 2007. Thus, the examiner has stated a purported advantage to the combination, but has not stated a reason or rational underpinning for how that purported advantage compels the *legal conclusion* of obviousness, as required by *KSR Int'l.* For this reason, the examiner’s statement is *conclusory*, which is in direct contravention to the requirements of *KSR Int'l.*

For example, the examiner stated a purported advantage and then assumed the legal conclusion of obviousness by assuming that the advantage must somehow compel the legal conclusion of obviousness. *A rational underpinning for the legal conclusion of obviousness is not the same as an advantage*; for example, one of ordinary skill would have to *recognize* the purported advantage, *have a reason to implement* the purported advantage, have the *technical skill* to combine the references, and also have *no reason to avoid* implementing the purported advantage in order to make the connection that one of ordinary skill would make the connection between the references in the first place. Additional logic would be required to state a compelling case for the *legal conclusion* of obviousness of the claim at issue; simply reciting an “advantage” is not enough under the standards of *KSR Int'l.* Therefore, under the standards of *KSR Int'l.*, the examiner failed to provide a rational underpinning to achieve the legal conclusion of obviousness, but rather provided only conclusory statements in stating rejection. Hence, under the standards of *KSR Int'l.*, the

examiner failed to state a *prima facie* obviousness rejection against claim 1 or any other claim in this grouping of claims.

### C. GROUND OF REJECTION 3 (CLAIM 5)

The third ground of rejection is the assertion that claim 5 is obvious under 35 U.S.C. § 103 over *Arnold* in view of *Poirier* and in further view of IBM Technical Disclosure Bulletin, Determining the Probability of Words in a String with a Word-Skipping Model, November 1, 1985 (hereinafter “*IBM Bulletin*”). Claim 5 is as follows:

The method of claim 3, wherein the language model is represented as a Hidden Markov Model.

The Final Office Action states the rejection of claim 5 as follows:

Claim 5:

*Arnold* and *Poirier* disclose the method of claim 3, but they do not disclose wherein the language model is represented as a Hidden Markov Model ([0023]).

The IBM Technical Disclosure Bulletin discloses a speech recognition method where the language model is "defined as a Markov source (a hidden Markov chain)" (page 2, lines 11-13).

Therefore it would have been obvious to one with ordinary skill in the art at the time of the invention to represent the language model as a Hidden Markov Model in *Arnold's* method because off the shelf Hidden Markov Model software was available therefore freeing *Arnold* from programming another model.

Final Office Action of July 17, 2007.

As previously shown, the proposed combination of *Arnold* and *Poirier* fails to teach the feature of claim 1, from which claim 5 depends. Claim 5 inherits the distinguishing feature of claim 1. Therefore, claim 5 is also distinguished from the combination of the cited references.

Additionally, the *IBM Bulletin* teaches at col. 2 lines 11-13 “the language model is defined as a Markov source (a hidden Markov chain).” The addition of the teaching of the *IBM Bulletin* fails to fill the gap in the teaching of *Arnold* and *Poirier*, and in particular that result caused by the opposing views of the two references. The *IBM Bulletin* fails to teach any feature missing from *Arnold* or *Poirier*, and the Final Office Action does not assert otherwise. Therefore, the teaching of

the proposed combination of the three references fails to teach or suggest all of the features of claim 5. Accordingly, no *prima facie* obviousness rejection can be made against claim 5.

**D. GROUND OF REJECTION 4 (CLAIMS 8-10, 15-16, AND 24)**

The fourth ground of rejection is the assertion that claims 8-10, 15-16, and 24 are obvious under 35 U.S.C. § 103 over *Arnold* in view of *Poirier* and in further view of *Lanning*, *Apparatus and Method for Optimizing Applications for Multiple Operational Environments or Modes*, U.S. Patent 5,787,285, August 15, 1995 (hereinafter “*Lanning*”). Claim 8 is representative of the group and is as follows:

The method of claim 7, further comprising:  
    profiling the decoder program to form a profile; and  
    determining whether the decoder program is optimized.

The Final Office Action asserts the proposed combination teaches the feature of claim 8 as follows:

Claim 8:

*Arnold* and *Poirier* disclose the method of claim 7, but they do not explicitly disclose profiling the decoder program to form a profile and determining whether the decoder program is optimized.

*Lanning* discloses a method of optimizing executable software where the code is profiled and optimized (“automated “profilers” to provide data to these optimizing compilers”, col. 1, lines 42-52).

It would have been obvious to one with ordinary skill in the art at the time the invention was made to profile and optimize the code in *Arnold* and *Poirier*’s method in order to “enhance the run-time performance of a piece of software” (*Lanning*, col. 1, lines 42-52).

Final Office Action of July 17, 2007.

The examiner failed to state a *prima facie* obviousness rejection against claim 8 because the proposed combination, including the addition of *Lanning*, fails to teach all of the features of claim 8. Specifically the proposed combination, considered as a whole, fails to teach the features of “profiling the decoder program to form a profile”; and “determining whether the decoder program is optimized.” The Final Office Action asserts otherwise, citing the following portion of *Lanning*:

Automated tools such as optimizing compilers are now commercially available to enhance the run-time performance of a piece of software. Recent technical papers have also disclosed the use of automated "profilers" to provide data to these optimizing compilers. In general, these profilers monitor the execution of a particular piece of software and gather information on the most likely paths of execution. The optimizing compiler then uses the information gathered by the profiler to recompile the source code in a more efficient manner.

*Lanning* col. 1 lines 42-52.

*Lanning* teaches providing a predefined set of inputs and data to the system for each operational environment or mode to be tested. *Lanning* further states that:

The first step in optimizing the code is to identify for each entry point, the basic blocks reachable from the entry point. That is, the basic blocks reachable from a first entry point are identified, along with the basic blocks reachable from a second entry point. In general, for each entry point, the code is analyzed to determine the basic blocks reachable from the entry point.

The next step is to determine the basic blocks which are executed within the various operating environments or operating modes at a frequency which exceeds the predetermined threshold value. This process is referred to as "profiling", and involves providing a predefined set of inputs and data referred to as a "profile" to the system for each operational environment or mode. The code is executed for each set of input conditions in the profile, while the code's run-time performance is monitored by the profiler. The profiler saves the run-time information in the profile data structure.

The profiling process is repeated until a response for each of the desired input conditions which correspond to a known operating environment or mode has been recorded. Each operational environment or mode for which the code will be optimized for includes a profile data structure containing predefined inputs and run-time information gathered while monitoring the response to the inputs.

Run-time characteristics, such as, the number of times each basic block was executed in a particular environment or mode are determined from the run-time information contained in the data structure.

Once the run-time information and characteristics have been determined, the code is partitioned into a plurality of "program segments" which each correspond to a particular environment or mode. Each program segment contains a plurality of basic blocks which are each executed at a frequency of execution above a predetermined threshold value within the operational

environment or mode. The remaining basic blocks are placed into a block of general purpose code which is accessible from each of the program segments.

*Lanning* col. 3 lines 1-32.

*Lanning* teaches in the referenced portion, and throughout the disclosure, the use of “profile” code to determine which code blocks should be replicated and which should be consolidated in performing optimization. *Lanning* describes an optimization process, and not a determination process prior to beginning the optimization process as claimed.

Further, *Lanning* teaches using “profiles” to test the code, in contrast to the claimed features of, “profiles the decoder program to form a profile” and then “determining whether the decoder program is optimized.” *Lanning* does not teach creating profiles from the code to be tested as is currently claimed. Therefore, *Lanning* teaches neither creating a profile from the code to be tested nor determining whether the code is optimized.

As the Final Office Action submits, neither *Arnold* nor *Poirier* teach the claimed features. As shown, *Lanning* fails to teach the claimed features. Therefore, the proposed combination, considered as a whole, fails to teach the claimed features. Therefore, the examiner failed to state a *prima facie* obviousness rejection against claim 8 or any other claim in this grouping of claims.

## **E. CONCLUSION**

As shown above, the Final Office Action has failed to state valid rejections against any of the claims. Therefore Applicants request that the Board of Patent Appeals and Interferences reverse the rejections. Additionally, Applicants request the Board to direct the Final Office Action to allow the claims.

/Theodore D. Fay III/  
Theodore D. Fay III  
Reg. No. 48,504  
**YEE & ASSOCIATES, P.C.**  
PO Box 802333  
Dallas, TX 75380  
(972) 385-8777

## CLAIMS APPENDIX

The text of the claims involved in the appeal is as follows:

1. A method, implemented in a data processing system, for generating task-specific code for pattern recognition, the method comprising:  
  
receiving task-specific input system data of a pattern recognition system; and  
  
generating task-specific code for the pattern recognition system based on the task-specific input system data, wherein the task-specific code includes computer language suitable for compilation.
2. The method of claim 1, wherein the pattern recognition system performs speech recognition.
3. The method of claim 2, wherein the task-specific input system data includes one of a language model, an acoustic model, a front-end for computing feature vectors, and information related to speaker adaptation.
4. The method of claim 3, wherein the acoustic model includes Gaussians.
5. The method of claim 3, wherein the language model is represented as a Hidden Markov Model.

6. The method of claim 3, wherein the acoustic model is represented as a Hidden Markov Model.
7. The method of claim 1, further comprising:  
compiling the task-specific code to form a decoder program.
8. The method of claim 7, further comprising:  
profiling the decoder program to form a profile; and  
determining whether the decoder program is optimized.
9. The method of claim 8, further comprising:  
responsive to the decoder program not being optimized, automatically modifying and  
recompiling the decoder program based on the profile.
10. The method of claim 7, wherein the step of compiling the task-specific code to form the decoder program includes:  
compiling the task-specific code in a plurality of parts to form a plurality of compiled parts, wherein the plurality of parts correspond to a plurality of modules of the pattern recognition system; and  
assembling the plurality of compiled parts before execution to form the decoder program.
11. A computer program product, stored in a recordable type medium, for generating task-specific code for pattern recognition, the computer program product comprising:

instructions for receiving task-specific input system data of a pattern recognition system;  
and

instructions for generating task-specific code for the pattern recognition system based on the task-specific input system data, wherein the task-specific code includes computer language suitable for compilation.

12. The computer program product of claim 11, wherein the pattern recognition system performs speech recognition.

13. The computer program product of claim 12, wherein the task-specific input system data includes one of a language model, an acoustic model, a front-end for computing feature vectors, and information related to speaker adaptation.

14. The computer program product of claim 11, further comprising:  
instructions for compiling the task-specific code to form a decoder program.

15. The computer program product of claim 14, further comprising:  
instructions for profiling the decoder program to form a profile; and  
instructions for determining whether the decoder program is optimized.

16. The computer program product of claim 15, further comprising:  
instructions, responsive to the decoder program not being optimized, for automatically modifying and recompiling the decoder program based on the profile.

21. An apparatus for generating task-specific code for pattern recognition, the apparatus comprising:

a bus;

a memory connected to the bus, wherein the memory contains computer readable instructions; and

a processor connected to the bus, wherein the processor executes the computer readable instructions to:

receive task-specific input system data of a pattern recognition system;

generate task-specific code for the pattern recognition system based on the task-specific input system data, wherein the task-specific code includes computer language suitable for compilation; and

compile the task-specific code to form a decoder program for the pattern recognition system.

22. The apparatus of claim 21, wherein the pattern recognition system performs speech recognition.

23. The apparatus of claim 22, wherein the task-specific input system data includes one of a language model, an acoustic model, a front-end for computing feature vectors, and information related to speaker adaptation.

24. The apparatus of claim 21, wherein the processor further executes instructions to:  
profile the decoder program to form a profile; and  
determine whether the decoder program is optimized.

## **EVIDENCE APPENDIX**

There is no evidence to be presented.

## **RELATED PROCEEDINGS APPENDIX**

There are no related proceedings.